EP18092I (2)

ELSEVIER

Computer Standards & Interfaces 18 (1997) 525–535

COMPUTER STANDARDS & INTERFACES

Relating the primitive hierarchy of the PREMO standard to the standard reference model for intelligent multimedia presentation systems

D.J. Duke [a,*], I. Herman [b,1], T. Rist [c,2], M. Wilson [d,3]

[a] University of York, Heslington, York YO1 5DD, UK

[b] Center for Mathematics and Computer Sciences (CWI), Kruislaan 413, Amsterdam 1098 SJ, Netherlands

[c] German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

[d] Council for the Central Laboratory to the Research Councils (CCLRC), Rutherford Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, UK

Abstract

The need for a suitable classification of media types arises for several reasons when building or comparing multimedia systems. Within an Intelligent Multimedia Presentation Systems (IMMPS), it is necessary to formulate and encode design knowledge for decision making on the appropriate medium in which to present information and for the generation of the presentation. It is also required in order to specify interfaces to and between system components which will be employed to run a generated presentation before the user's eyes. This task is reflected in the Standard Reference Model (SRM, see this volume) for IMMPS by the Presentation Display Layer. However, the SRM does not instantiate this layer in detail, but instead refers to the Presentation Environment for Multimedia Objects (PREMO) ISO/IEC standard which provides a reference model for a presentation runtime environment for multimedia. PREMO already contains a set of basic structures, the so-called PREMO Primitive Hierarchy, to describe different media types. Thus the question arises, as to how far the PREMO Primitive Hierarchy could serve as a media classification for the SRM in general. In particular, this would support consistency between the design and presentation layers of the SRM if PREMO were used to instantiate the presentation layer. In the current paper, we first point to a number of typical problems with generating classifications of media types. We then provide a brief introduction to PREMO and its Primitive Hierarchy. Finally, the benefits and costs of using the PREMO Primitive Hierarchy for the SRM are discussed. © 1997 Elsevier Science B.V.

Keywords: Multimedia; Multimedia modelling; Knowledge representation; PREMO; Geometric primitives; Standards

1. Introduction

Bordegoni et al. [1] present a Standard Reference Model (SRM) for Intelligent Multimedia Presentation Systems (IMMPS). In outline, this model consists of a layered pipeline to generate presentations, and which can call on knowledge servers to control its decisions. The Content Layer contains a Media

* Corresponding author. E-mail: duke@minster.york.ac.uk

[1] E-mail: ivan@cwi.nl

[2] E-mail: rist@acm.org

[3] E-mail: mdw@inf.rl.ac.uk

PII S0920-5489(97)00017-2

The choice of a certain perspective is often a choice between cognitive adequateness on the one hand, and engineering pragmatism on the other hand. Consider, e.g., the choice of a color model in a media hierarchy. The well-known RGB model is based on hardware design, as well as human biological considerations (the human visual system is also based on mixing various primary colors, just as the RGB monitors do). In other words, the RGB model is very well adapted to (computing) systems. On the other hand, the same RGB system is very counterintuitive; anybody, who has ever tried to set a specific color on a display, can witness this. The 'perceptual' counterpart is the HSV system, which is user-oriented, being based on the intuitive appeal of the artist's tint, shade, and tone. The price to pay when using this model is that an extra transformation from the HSV values to the RGB system becomes necessary (to control the underlying hardware), and that the classic color-based algorithms of synthetic graphics (shading, etc.) become more complicated (see Ref. [8]).

With regard to the formulation of design knowledge, a media classification certainly must not ignore the human-centered perspective. To avoid misuse and underuse of media the yardstick of any assessment must involve the human user (see also Ref. [9]). On the other hand, engineering pragmatism has taken into account when striving for a specification of interfaces between layers and components of the SRM.

2.2. *Conceptualization: objects vs. attributes and methods*

When building a model of a domain, it is not always apparent what should become an object, and what should be considered as an attribute to an object or a relation between objects. In general, one may say, the more abstract the domain, the more difficult the decision. Graph theory even shows that in some cases a structurally equivalent model can be obtained simply through dualization (i.e., objects become relations and vice versa). The conceptualization issue is especially relevant, if one strives for an object-oriented classification such as the PREMO Primitive Hierarchy.

The practice and pragmatics of OO programming can lead to different classifications, too. A classic example is as follows. A fundamental question that must be addressed within any object-oriented graphics or multimedia system concerns the allocation of fundamental behavior, such as transformations and rendering, to object types within an API. Two quite distinct approaches emerge. The first is to attach behavior to the object types that are affected by that behavior. For example, geometric objects and other kinds of presentable media data can be defined with a 'render' method, with the interpretation that such an object can be requested to produce a rendering of itself. Such an approach can be extended to collections of presentable objects, and fits well with the concept of an object as a container for data along with the operations that manipulate that data. The second approach is to define objects whose principle purpose is to act as information processors, and which receive the data that they operate on as parameters to operation requests or through some other communication mechanism. In this case, a 'renderer' object would receive presentable objects as input through some interface, and produce a rendering of those objects via some output mechanism.

With regard to the issue of conceptualization, one may conclude that any classification of media types must leave sufficient freedom to allow designers the choice of their design philosophy.

2.3. *Structure of classification: flat list vs. minimal hierarchy*

A good classification should be organized in a way that reflects the relevant differences between the essential properties of the classified items, and at the same time, removes redundancies. Choosing the optimal set of primitives for media objects, however, remains a difficult endeavor as long as new multimedia input and output devices are appearing from one day to the next. That is, the development of a classification that is 'complete' in any useful sense is highly problematic. Therefore, a useful media classification must also allow the systematic integration of new media, preferably without resort to the most naive approach to classification, which is simply an open-ended flat list of media types.

3.2. *PREMO is aimed at a multimedia presentation*

Whereas earlier SC24 standards concentrated on either synthetic graphics or image processing systems. Multimedia is considered here in a very general sense; high-level virtual reality environments, which mix real-time 3D rendering techniques with sound, video, or even tactile feedback, and their effects, are, e.g., within the scope of PREMO.

3.3. *PREMO is object-oriented*

This means that, through standard object-oriented techniques, a PREMO implementation becomes extensible and configurable. Object-oriented technology also provides a framework to describe distribution in a consistent manner.

3.4. *PREMO is a framework*

This means that the PREMO specification does not provide all the possible object types for making graphics or multimedia. Instead, PREMO provides a general programming framework, a sort of middleware, where various organizations or applications may plug in their own specialized objects with specific behavior. The goal is to define those object types which are at the basis of any multimedia development environment, thereby ensuring interoperability.

At the time of writing (Summer 1997), PREMO is in DIS stage; this means that, on the one hand, its technical content is now more or less final and, on the other hand, that it will become an official ISO Standard in 1998.

A precise object model constitutes a major part of PREMO. The object model is fairly traditional, and is based on the concepts of sub-typing and inheritance. It is also very pragmatic in the sense that it includes, for efficiency reasons, the notion of non-object (data) types, as is the case with a number of object-oriented languages, such as C + + or Java, and in contrast to 'pure' object-oriented models such as SmallTalk. The PREMO object model originates from the object model developed by the OMG consortium for distributed objects, but some aspects of the OMG model have been adapted to the needs of PREMO. The model has also undergone a thorough formal specification process (see Ref. [11]). Note that here is a strong emphasis in PREMO to make it well adapted to distributed environments; this emphasis also directed some of the design decisions reflected below.

4. The PREMO primitive hierarchy

PREMO is concerned with the presentation of multimedia information, and in allowing different renderers to inter-operate within a potentially distributed system. Also, it was an important design requirement of PREMO to allow for extensibility, i.e., that either applications or other, standardized, components would add their own set of primitives to the PREMO framework. For these reasons, the PREMO standard does not attempt to define the structure of primitives to the same level of detail as found, e.g., in graphics standards such as GKS and PHIGS (e.g., see Ref. [12]). Instead, the approach in PREMO is to provide a general, extensible framework that provides a uniform basis for deriving primitive sets appropriate to specific application or renderer technologies. In general, modellers or renderers may use specific techniques, such as Constructive Solid Geometry for a particular range of applications. Such techniques may require an enriched set of basic primitives. The aim of the PREMO primitive hierarchy is to provide a minimal, common vocabulary of structures that can be extended as needed, either by applications using PREMO, or by other standard components.

Referring to one of the issues cited in Section 3. PREMO has deliberately avoided adding explicit procedural 'behavior' to the primitive objects, explicitly separating media processors such as renderer objects from the primitives. One of the main reasons is the fact that PREMO should operate in a distributed model, where one model or data set may be rendered by several processes working in parallel at various locations. It is difficult to see how this can be realized efficiently in an architecture in which each model object renders itself.

Fig. 1 shows the subtype hierarchy of the PREMO primitives. In PREMO, the concept of primitive encompasses the description of both structure and appearance. At the top level, PREMO distinguishes between seven kinds of primitives, which will be described in somewhat more detail in Fig. 1.

ants of media (e.g., their own 'view' of surfaces, point data set, haptic properties, text descriptions, etc.). Additional kinds of form primitives may be added in future to include other categories such as olfactory and taste.

4.2. Captured primitives

A captured primitive contains a reference to a source of raw data encoded in some standard format such as JPEG, MPEG, MIDI, or VRML. This data may happen to be recorded, or live. The detailed specification of this primitive refers to another part of PREMO, called the Multimedia Systems Services, which provides an abstraction for the various (multimedia) virtual devices which may produce such raw data [4].

4.3. Modifier primitives

The primitives in this category have no perceivable representation by themselves. Instead, they carry information that affects the presentation of other primitives. Examples include visual effects (color and texture), geometric transformations, or audio effects. The modifiers have been grouped to reflect the kind of effect that they produce, and the kind of primitives to which they can be applied. PREMO does not describe the order in which modifiers are applied, and whether or not they are accumulative or override previous modifications. The reason for this non-commitment is that applications may realize graphical rendering through existing systems and standards, within which the order and scope of modifications within the rendering pipeline or scene structures varies widely.

4.4. Reference primitives

A reference primitive introduces a link to a structured primitive defined in some other part of the hierarchy. It contains a single name-valued attribute, label, that is intended to be matched against a similar name in a primitive structure.

4.5. Structured primitives

Form, captured, and modifier primitives can be viewed as atomic units of information that determine or affect the presentation of a multimedia system. Such systems, however, need to define and manipulate collections of primitives, both to represent large-scale or application-specific structures, and to coordinate the presentation of primitives over time. These two roles are somewhat different, and are reflected in PREMO by two object types that encapsulate a collection of primitives. This collection may itself include structured primitives, allowing the construction of hierarchical structures.

Aggregates allow a number of primitives to be combined into a structure without imposing any interpretation on the meaning of such a collection. They provide a facility for building larger-scale primitives and also allow an application to group semantically related primitives into single units that can be named. Application or other standard components may impose a particular view of structuring (e.g., Directed Acyclic Graphs). Aggregates also have a naming mechanism, whereby primitives can be labelled by a name built from a sequence of strings. This name can be used by reference primitives, and/or various selection mechanisms.

Time and temporal extent are fundamental to multimedia presentation and in general a multimedia system will contain a number of primitives which need to be synchronized in time. Although time could arguably be treated in a way similar to that used for spatial coordinates, most multimedia systems will typically treat time in a specialized way, to support the realization of various time-related constraints, synchronization, etc. The *Time Composite* object of PREMO has been introduced to structure primitives in the time domain. It contains a sequence of component primitives, inherited from the *Structured* primitive object type, that defines the content of the composite. The object also contains various attributes which make it possible to monitor and control the timing of the composite as a whole. These include duration, start and end time 'buffers' that provide flexibility in coordinating the presentation of multiple *Time Composite* objects, and an event monitor through which external objects can be informed of, e.g., the progress of a renderer in processing the object. PREMO defines three specific subtypes of *Time Composite*:

- Sequential time composite, in which the component primitives are presented in sequential order;

be negative in this case. This is not surprising since the PREMO Primitive Hierarchy is a purely system-centered classification and does not reflect a human-centered perspective at all. To be more precise, some of the criteria used by the PREMO designers for distinguishing their primitives are not relevant for the formulation of design knowledge. Vice-versa, there are criteria (e.g., the user's cognitive effort for processing media objects of a certain type, applicability constraints, etc.) which are not considered in PREMO but which cannot be neglected when defining a suitable classification of media types that would allow to formulate design knowledge in an intuitive manner. Consider, e.g., the task of media allocation, which is to select from available presentation media the one which can most effectively convey a given information. Whether or not a certain medium fulfills the requirement of being effective can only be answered by relating the properties of media to the capacities and peculiarities of human perceptual processes. The primitives in the PREMO hierarchy do not establish such a relation. The distinction drawn by the PREMO primitives can even lead to more complicated design rules. For example, if the PREMO primitives were used to formulate a naive rule such as 'use graphics for localization tasks', then it would be necessary to replace in the rule the term 'use graphics' by the less intuitive expression 'use either Geometric, or Captured, or Structured'. However, for the consumer of the presentation it doesn't make any difference whether the graphics has been generated by the system, whether the system presents a 'captured' graphics, or whether the presented graphics has been composed of some generated and some captured parts.

6. Conclusions

In this paper, we have discussed some of the typical problems with trying to establish a classification for media types. We have presented the Primitive Hierarchy of the PREMO standard for multimedia runtime environments. Being thoroughly designed to capture the broad array of available media, this hierarchy is the best system-centered definition of media types we currently have. The question was raised, whether the PREMO primitive hierarchy would be a suitable adjunct to the SRM for IMMPS [1]. One issue in the context of the SRM is the specification of interfaces between layers and components of layers. For this purpose, the PREMO primitive hierarchy has been considered useful. Moreover, adopting this classification would facilitate the instantiation of SRM's Presentation Display Layer with PREMO as it is already suggested by the proposers of the SRM [1].

For the purpose of formulating design knowledge, however, the PREMO Primitive Hierarchy appears to be too much system-centered. This should not be understood as a criticism to the designers of PREMO, since, for them, the formulation of design knowledge was never an issue. One may rather conclude that further research is needed in order to establish an ideal media classification which merges both the human-centered perspective and the system-centered perspective. Since the device-related descriptions of media, and those required to capture rules of thumb, differ so greatly, complexity is required at one stage of the process: either to map from the rules of thumb to a device-based description, which would constrain rule structure but improve runtime efficiency, or from a user-centered description in which rules of thumb are easily expressed to the devices within the SRM at runtime. Similarities were suggested earlier between this attempt to standardize media taxonomies the problem of describing color (this latter has been the subject of research for a long time). The usual solution in the case of color was to use one standard (RGB) which easily mapped to the implementation of systems, and could be used to describe them; a second standard (HSV) was introduced as a user-oriented description, while a third (CIE) was used to map between the two, incorporating both psychological and physical aspects of color. (Indeed, the CIE color model addresses the problem of device independent specification of color by placing the monitor phosphors and white points precisely in color space.) A similar solution may be required in order to formulate design knowledge of media, with one description, such as PREMO, to be used for the generation environment, and another, which is user-centered, to be used to elicit design knowledge and integrate empirical findings with rules of thumb (perhaps based on the proposals of Bernsen [5]), while a

http://www.cwi.nl/ivan.Dr. Thomas Rist is a senior researcher at the Department Intelligent User Interfaces of the German Research Center for Artificial Intelligence (DFKI). Dr. Rist studied computer science and mathematics at the University of the Saarland. He undertook his doctorate in the area of knowledge-based graphics generation and received the degree from University of the Saarland. He was involved in various projects funded by the German Ministry for Education and Research, industries, and the EU. One of these projects, WIP, was a winner of a 1995 Information Technology Award (ITEA). As a Senior Scientist, he was also in charge of management tasks in a number of industrial projects, and is currently coordinator of the European LTR Project Magic Lounge within the i3 initiative on Intelligent Information Interfaces. His areas of interest and experience include: multimedia/multimodal communication, intelligent user interfaces, automated presentation systems, knowledge-based graphics generation, life-like characters, and user modelling. Dr. Rist has served as a programme committee member for national and international conferences and workshops, and was on the organization board of a number of workshops on multimedia systems. In 1995, Dr. Rist became a member of the ERCIM Computer Graphics Network Task 2 Group.

Michael Wilson obtained a bachelors degree in Experimental Psychology from the University of Sussex, and a doctorate in Psycholinguistics from the University of Cambridge. After 1983, he studied how users learn to use windowing systems on a project funded by IBM at the MRC Applied Psychology Unit, Cambridge. Since 1986, he has worked at the Rutherford Appleton Laboratory researching intelligent user interfaces incorporating multiple interaction modes, automatic generation of presentations, ontology-based information retrieval, knowledge acquisition, multimedia information retrieval and presentation, and conversational interaction. He is a member of the BCS and ACM. and has served on the programme committees of many international conferences, and several journal editorial boards on HCI and AI. He has acted as a research programme advisor and reviewer for UK and European research programmes, and is currently coordinator of the UK EPSRC programme on Multimedia Networking Applications.